CCID5: An implementation of the BBR Congestion Control algorithm for DCCP and its impact over multi-path scenarios

Nathalie Romo Moreno, Markus Amend, Anna Brunstrom,
Andreas Kassler, Veselin Racoceciv

Introduction

Design and implementation

Evaluation

Conclusion and Future work

Introduction

Background and Objective

Multi-connectivity and Hybrid Access technologies	Current Standards	Multi-path solution for non-reliable traffic	Congestion Control algorithms
- Simultaneous use of heterogeneous networks - Capacity aggregation - Reliability - Flexible resource management	- Broadband Forum standard for Hybrid Access - Access Traffic Steering Switching Splitting (ATSSS) specified first in 3GPP Rel. 16 -> MPTCP -MPTCP -> solution for traffic splitting -> reliable and in order transport of traffic	-**MP-DCCP protocol** -> extends DCCP to support multi-path sessions -> Unreliable transport of data -Connection-oriented and congestion controlled -**MP-DCCP framework** -> MP-DCCP protocol + Scheduling + reordering + virtual network interfaces	-Key factor to keep a low latency difference among paths -> improve end to end performance -DCCP CC algorithms-> CCID (Congestion Control Identifier) - Available CCIDs :CCID2, CCID3 and CCID4 . All based on existing TCP loss-based CC algorithms

Improve the MP-DCCP framework by extending DCCP with a new congestion control algorithm

-CC algorithm selected: **BBR (Bottleneck Bandwidth Round Trip propagation time)**

-Initially developed for TCP

-Low latency

-High throughput

- MP-DCCP (protocol) -> https://datatracker.ietf.org/doc/html/draft-amend-tsvwg-multipath-dccp-05
- MP-DCCP (Framework) -> https://datatracker.ietf.org/doc/html/draft-amend-tsvwg-multipath-framework-mpdccp-01

Design and implementation

BBR - BOTTLENECK BANDWIDTH ROUNDTRIP PROP.

Algorithm description

Optimal point of operation

- Inflight = BDP = BtlBw*RTProp
- Bottleneck packet arrival must match the BtlBw

Input parameters

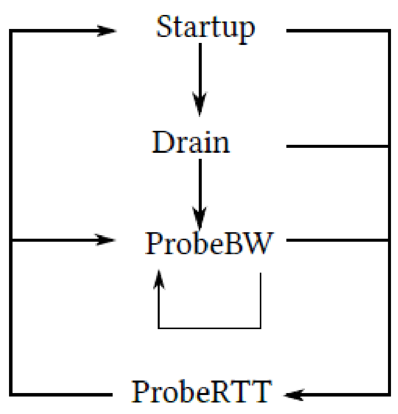
- BtlBw and RTProp

Control parameters

- Congestion window (cwnd), Pacing rate and send_quantum

State Machine

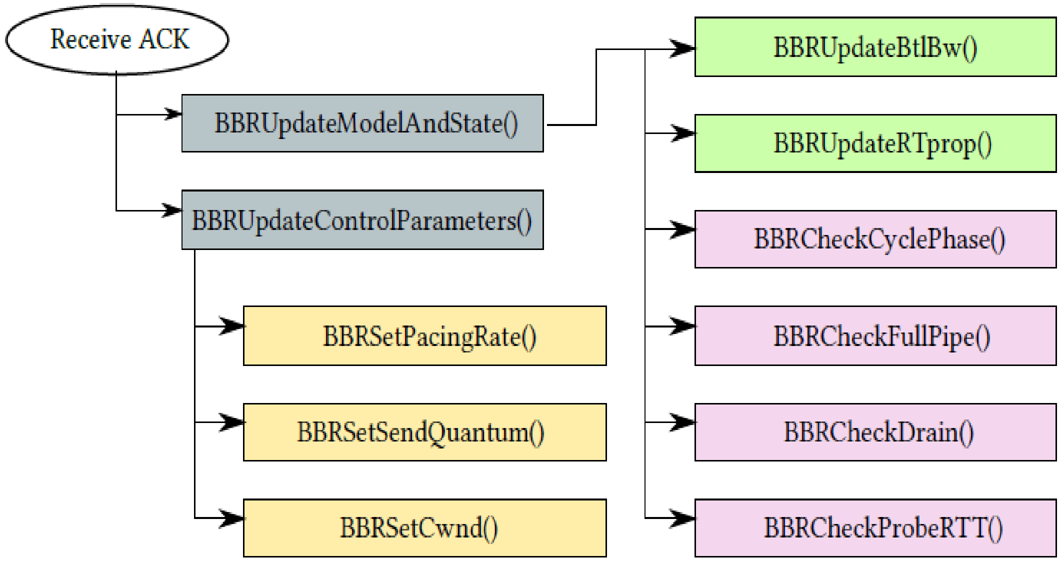
- Startup: sending rate -> increase rapidly
- Drain -> sending rate will be reduced
- Probe BW -> amount of data inflight slightly increased
- Probe RTT-> data inflight reduced



TCP vs DCCP implementation

General considerations

- BBR v1->CCID5 (for DCCP) -> Within the Linux kernel 4.14 -> available as open source. https://github.com/telekom/mp-dccp/blob/master/net/dccp/ccids/ccid5.c
- Guidelines and pseudo-code from IEFT drafts -> https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-00, https://datatracker.ietf.org/doc/html/draft-cheng-iccrg-delivery-rate-estimation-00
- Reuse and adapt code from BBR TCP implementation in the Linux kernel



DCCP implementation challenges

Acknowledgment

- format of the ACK packets, the timing of their generation, and how they are congestion controlled.
- Definitions taken from CCID2: ACK vector and ACK ratio

Additional functions

- Tracking packets in flight, sending and arrival times
- Application limited period

Integration with Multiptah framework

General description

Modular scheduler

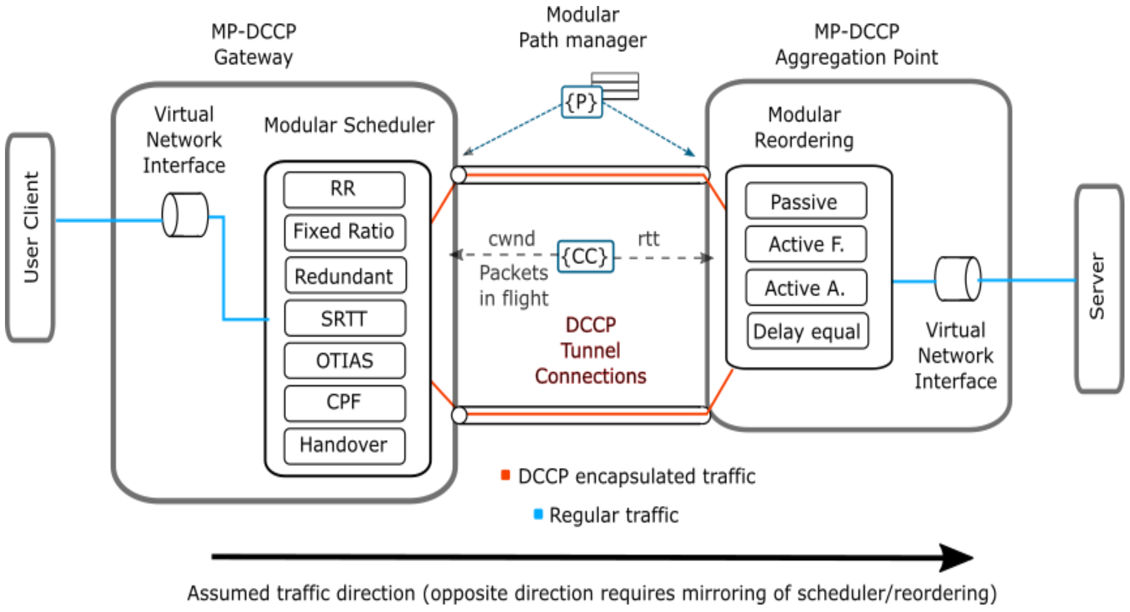
- CPF (Cheapest Pipe First) : Allocates packets based on a predefined path priority. Paths fully congested are skipped from the selection.
- RR (Round Robin) Alternates packet-sending through all the available paths. Paths fully congested are skipped from the selection.

Modular reordering

- Active-Fixed: Reads packet sequence numbers to verify in order arrival. When a gap is detected, a buffer is used to store received packets until the missing one(s) arrive, or a fixed timer expires.

Information needed from CC algorithm

- cwnd and Packets in flight -> used by the schedulers
- RTT estimation -> used by some reordering algorithms



Evaluation and results

Topology and metrics

Topology description

Virtual environment

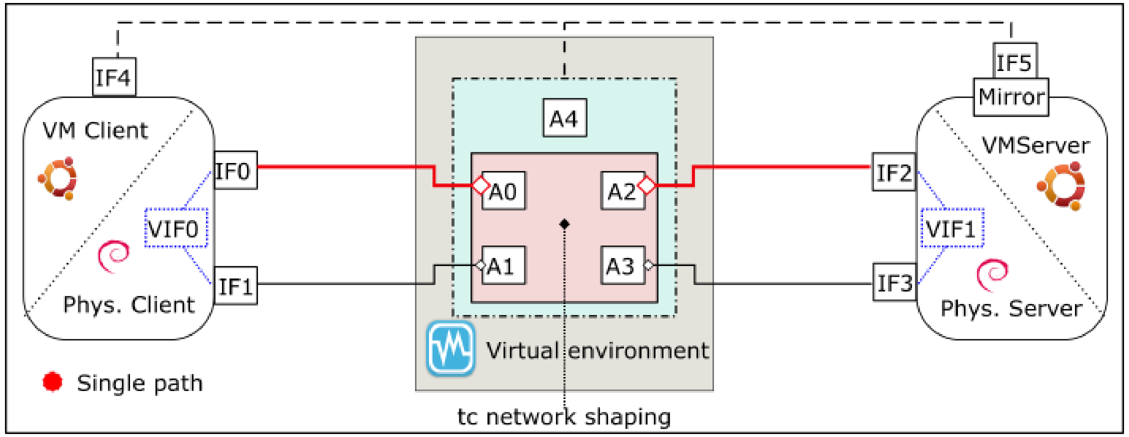
- host machine running Ubuntu 16.04.6
- Oracle VM VirtualBox version 5.1.38.
- Network -> host only adapters (denoted
- VMs run the same Ubuntu version as the host but with the Linux Kernel version 4.14.114 enhanced with the CCID5 module.
- Red path -> single path scenario

Physical environment

- PC engines APU boards running Debian 9.3 -> directly connected.

Metric measurement

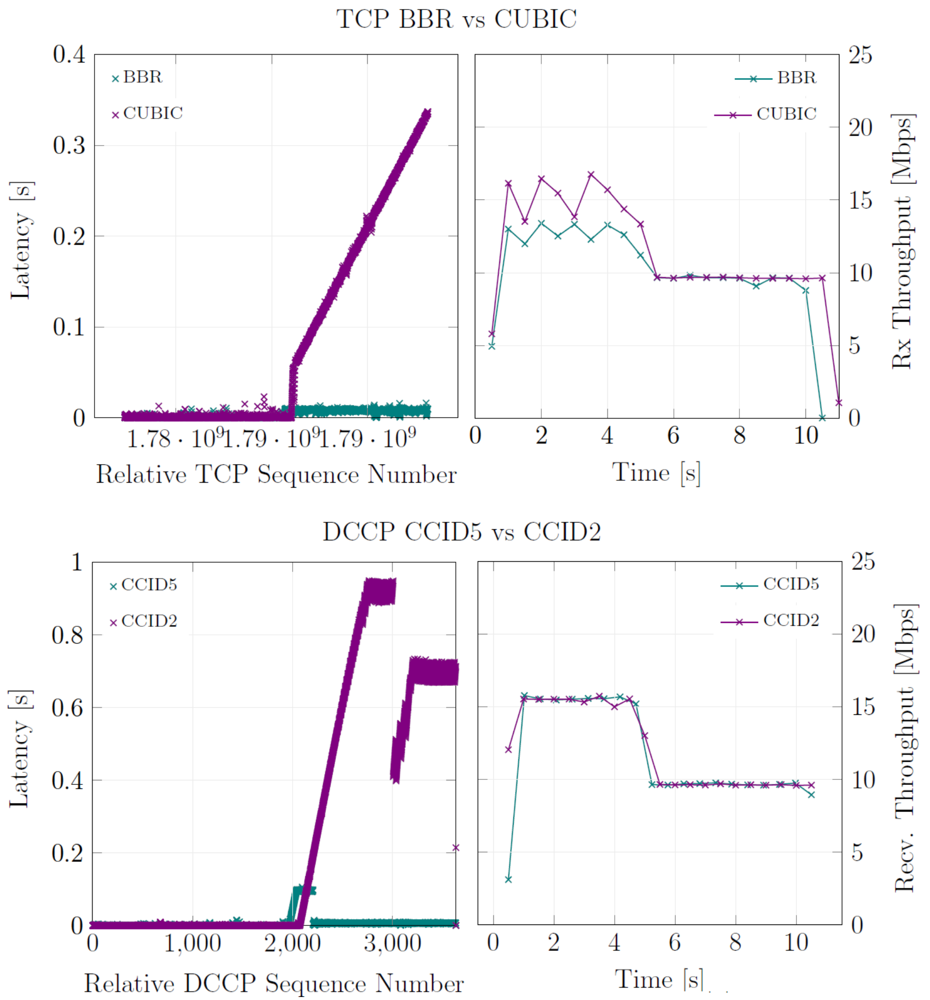
- End to end latency -> Traffic is Mirrored from server to client. Outgoing and incoming (mirrored) traffic are captured to measure timestamp difference per packet.
- Received throughput-> Incoming traffic at server side is captured .Wireshark's IP graph tool is used to plot throughput over time.



Single Path scenario

Test parameters

Traffic Type	CC Algorithm	Tx Rate	Path BW	Duration
TCP	CUBIC	15Mbps	1G for $t < 5s$ 10Mbps for $t \geq 5s$ (*tc*)	10s
	BBR			
DCCP	CCID2			
	CCID5			



Result analysis

- After Bw change , BBR and CCID5 maintain a low latency and a throughput equivalent to the available path BW
- CCID2 and CUBIC fill the buffer created by the *tc* rule causing higher latencies

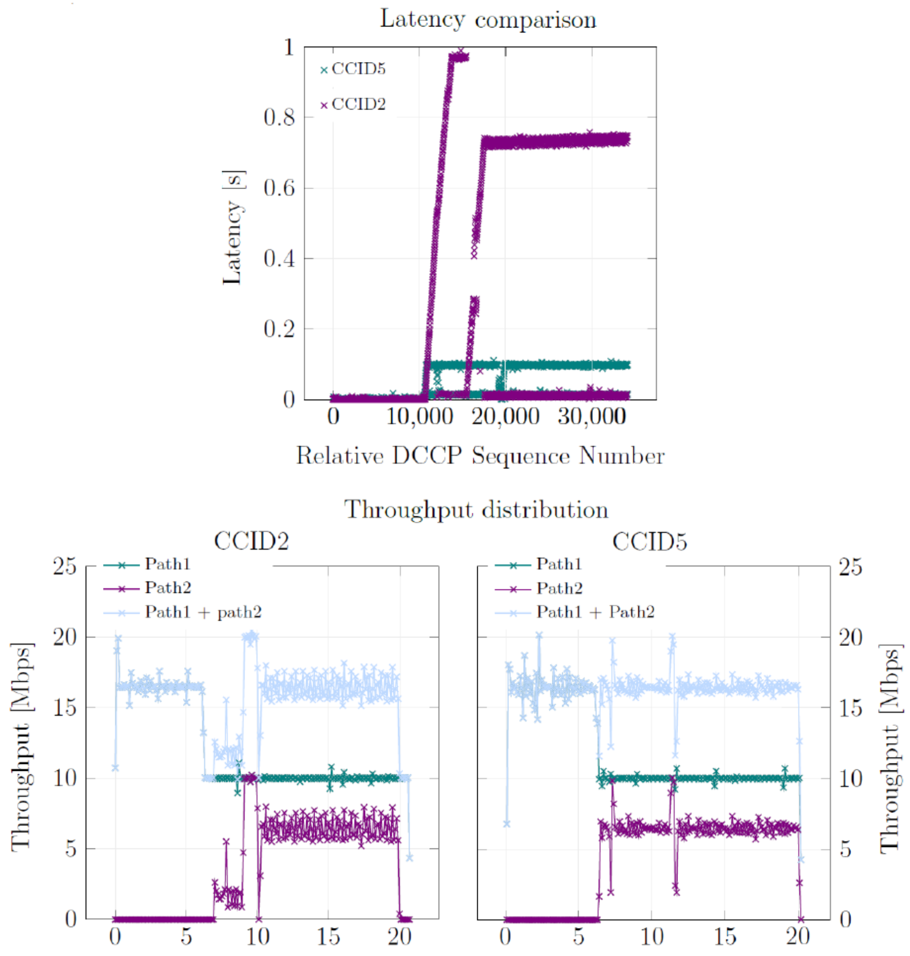
Multi-path scenario UDP

Test parameters

Sched.	*Reordering*	Tx Rate	Path BW	Duration
CPF	Active Fixed	15Mbps	1G for $t < 5s$ 10Mbps for $t \geq 5s$	20s

Result analysis

- First seconds -> CPF selects one path
- After bandwidth change CCID5 estimates available BW -> congestion status is informed to the scheduler and the traffic is distributed across both paths
- CCID2 takes longer to detect congestion (loss) and inform to the schedules. The prioritized path remains fully congested leading to higher latencies



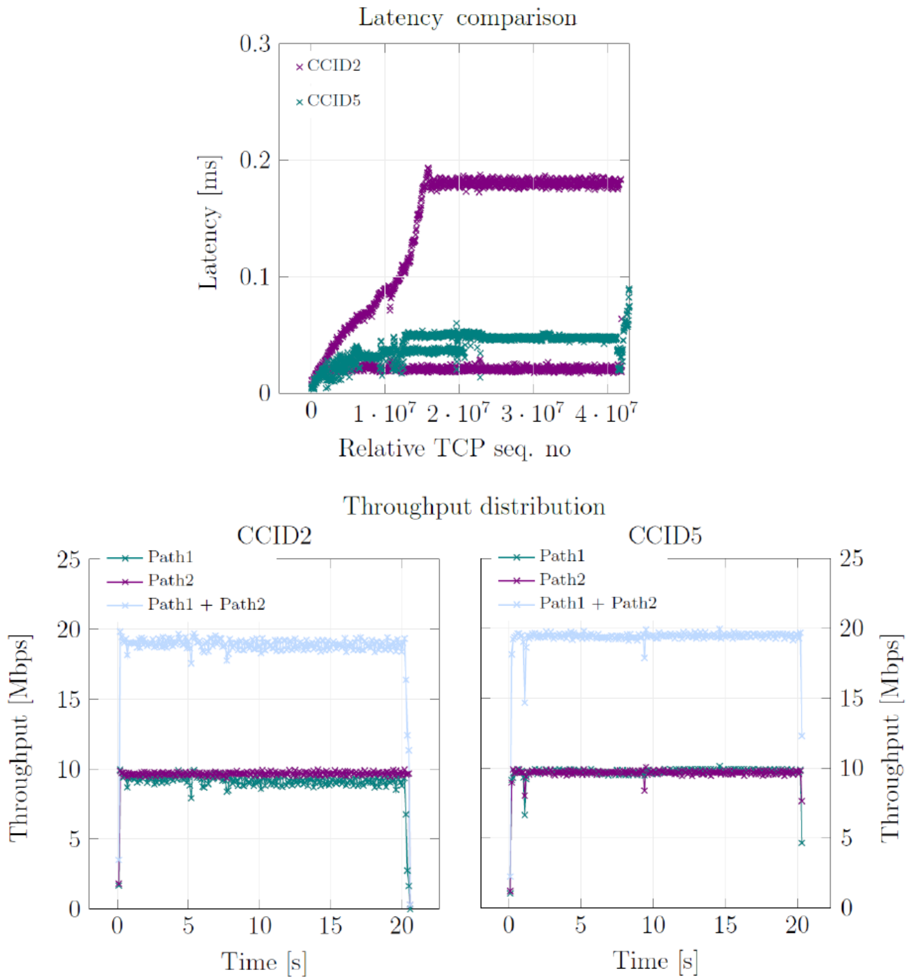
Multi-path scenario TCP

Test parameters

Sched.	*Reordering*	Tx Rate	Path BW	Duration
RR	Active Fixed	NA	10Mbps (set via ethtool)	20s

Result analysis

- Similar performance in terms of throughput for both CCIDs
- CCID5 shows lower latencies in comparison with CCID2



Conclusion and Future work

Conclusion and Future work

Conclusion

- CCID5 has clear benefits with respect to CCID2 in the multipath scenario –> reduces latency avoiding head of line blocking and minimizing the reordering effort
- We implement BBR congestion control as a new CCID in the DCCP protocol, proving that the mechanisms proposed by BBR to characterize the network path and update the sending behavior, accordingly, are suitable for DCCP.
- We integrated our approach into a multi-access framework that can be applied to the 5G ATSSS context or to the Hybrid Access scenario

Future work

- Extensive testing
  - concurrent flows, path delay and packet loss
- BBRv2
- Standardization